Basic Networking

What Is The Purpose Of Networking Connection?

The purpose is to establish a channel of comunication between client and server for sending and receiving data. You can manage this information, as generates a XML file, JSON file, etc.

Features

- Cross-platform HTTP requests
- Multi-platform TCP client and server socket support with configurable settings
- Optimized TCP client and server settings for low-latency
- Cross-platform browser access.

Interfaces

- **Socket** → is an interface that provides you with the remote socket address, connection state, and a java.io.InputStream and java.io.OutputStream to work with the socket.

- **ServerSocket** → is an interface used to create TCP server sockets. It provides the standard **accept()** method to get a TCP client that connected.

Classes

- **SocketHints** → is a class used to configure TCP Cliente

- **ServerSocketHints** → is a class used to configure Server Socket

Create A Server Socket

```
ServerSocket server = Gdx.net.newServerSocket(Protocol.TCP, localhost,
6066, new ServerSocketHints());
```

We can change the server socket configuration creating a new ServerSocketHints and editing its properties.

E.g: If we need to change the timeout connection, for example, the new way is the following:

```
ServerSocketHints hints = new ServerSocketHints();
hints.acceptTimeout = 120;     // The time is in milliseconds.

ServerSocket server = new Gdx.net.newServerSocket(Protocol.TCP, 6066,
hints);
```

Create A Client Socket

In the part of client, you need to make the following code:

```
Socket socket = Gdx.net.newClientSocket(Protocol.TCP, 123.456.789, 6066,
new SocketHints());
```

We can change the client socket configuration creating a new SocketHints and editing its properties.

E.g: The same as change the timeout connection in the part of server, we make the following for the client:

```
SocketHints socketHints = new SocketHints();
socketHints.connectTimeout = 3000;    // The time is in milliseconds.

Socket socket = Gdx.net.newClientSocket(Protocol.TCP, 123.456.789, 6066,
socketHints);
```